УДК 004.424

С.С. Синельников, В.А. Резников
Государственный университет информатики и искусственного интеллекта,
г. Донецк, Украина

Разработка структуры данных для задачи поиска методом Ньютона

Рассмотрены проблемы обобщения существующих методов поиска данных, применения динамических структур данных к задаче поиска, применения метода Ньютона. Показан комплексный подход к решению задачи поиска. Разработан метод поиска данных, основанный на применении динамических структур данных и разбиении массива на группы элементов.

Введение

В работе [1] показана связь численных методов поиска нуля функции [2-5] с задачей поиска индекса элемента массива по его ключу [6-8]. Было доказано, что метод Ньютона дает решение задачи поиска индекса, но при этом требует, чтобы элементы исходного массива m удовлетворяли следующим условиям:

$$m[i] <= m[i+1], \quad (1)$$

$$m[i+1] – m[i] <= m[i+2] – m[i+1]. \quad (2)$$

Условие (1) выполняется, так как массив заведомо отсортирован по возрастанию. Условие (2) не выполняется в общем случае, что говорит о том, что метод Ньютона является достаточно узконаправленным. Метод Ньютона при выполнении условий (2) обладает высокой скоростью поиска – порядка loglogN. По сравнению со скоростью бинарного метода поиска – logN, данный метод обладает лучшими характеристиками, а в отличие от метода хорд (который обладает такой же скоростью поиска) – имеет более простой и быстрый алгоритм проверки условий, необходимых для применения этого метода.

Методы хорд и Ньютона выполняют наименьшее количество сравнений из всех известных методов на сегодняшний день, но требуют выполнения определенных условий для взаимного расположения данных, прежде чем применять их для задачи поиска, поэтому данные методы необходимо развивать и обобщать.

В данной статье предлагается динамическая структура данных, которая позволит применить сильные стороны алгоритма Ньютона и избавиться от недостатков (условие (2)), которые не позволяют применять данный метод для любого набора данных.

Алгоритм поиска методом Ньютона

Для поиска данных, удовлетворяющих условиям (1) и (2), в массиве можно использовать алгоритм, блок-схема которого представлена на рис. 1.

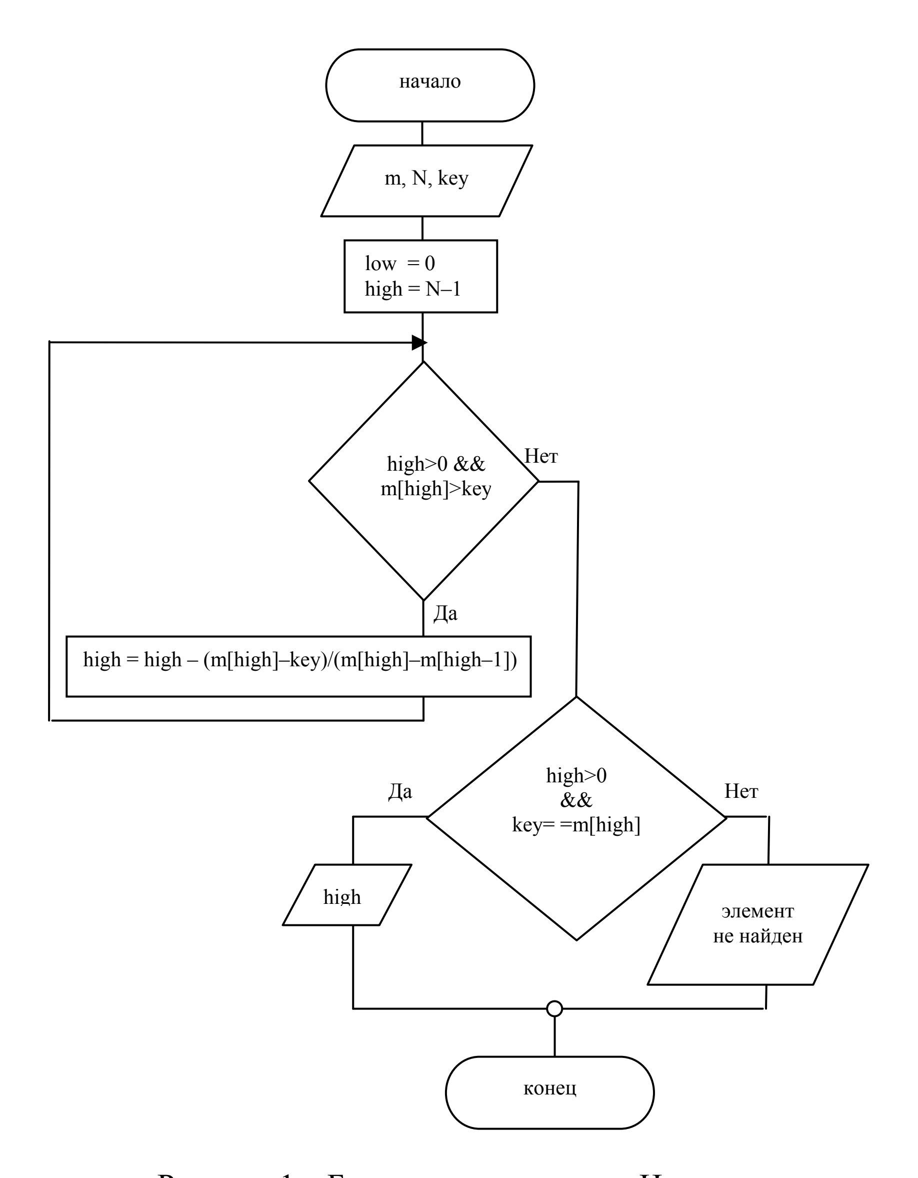


Рисунок 1 – Блок-схема алгоритма Ньютона

Разработка динамической структуры данных для поиска методом Ньютона

При использовании динамических структур данных, отличных от одномерных массивов, формулировка задачи поиска несколько изменяется, в соответствии с которой после поиска элемента возвращается его физический адрес в памяти (или ссылка на него), а не индекс (как было в случае с одномерным массивом). Таким образом, задача поиска элемента заключается в нахождении в организованной структуре данных адреса искомого элемента. Причем результатом поиска может служить как адрес, так и ссылка [9] на искомый элемент. В случае, если элемент не найден, то возвращается нулевой адрес (NULL).

Рассмотрим одномерный отсортированный массив m с размерностью N (рис. 2). Данный массив удовлетворяет условию (1) и не удовлетворяет условию (2), в связи с чем метод Ньютона не применим.

m[0]	m[1]	…	m[N–1]

Рисунок 2 – Массив m размерностью N

Выберем в группу G_{00} из массива m возможные KOL элементов $m[k_0]$, $m[k_1]$, … $m[k_i]$, $m[k_{i+1}]$ … $m[k_{KOL-1}]$, где для каждого элемента выполняются условия (3):

$$m[k_{i+1}] - m[k_i] <= m[k_{i+2}] - m[k_{i+1}];$$
$$k_0 = 0,\ k_1 = 1; \qquad (3)$$
$$k_i < k_{i+1},\ i = 0, KOL - 1.$$

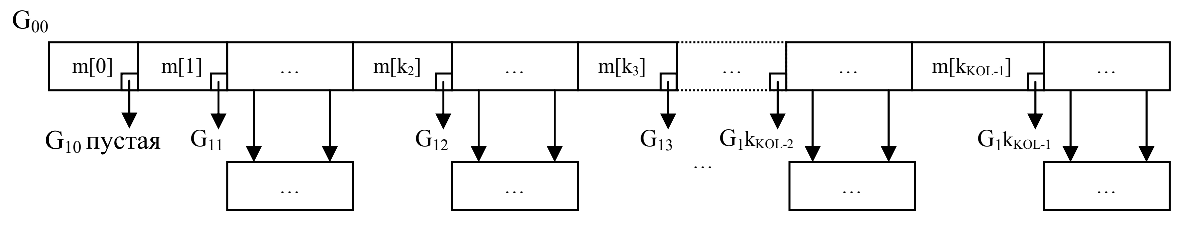


Рисунок 3 – Разбиение массива m на группы с формированием динамической структуры данных (первый шаг)

Для полученного массива G_{00} выполним переиндексацию массива, начиная с нуля. Для данного массива можно применять поиск методом Ньютона, так как условия (1) и (2) выполняются.

С оставшимися элементами, которые не вошли в массив группы G_{00}, необходимо выполнить следующие действия:

1) сформировать группы G_{10}, G_{11}, …, G_{1i}, $G_{1,i+1}$, …, $G_{1,KOL-1}$ элементов, которые не вошли в G_{00} и находились соответственно между элементами $m[k_i]$ и $m[k_{i+1}]$. Если же группа не содержит элементов, то будем называть ее пустой (NULL);

2) переиндексировать элементы для каждой полученной группы, начиная с нуля;

3) для массива каждой не пустой группы повторить действия, аналогичные действиям с исходным массивом m;

4) выполнять создание структуры до тех пор, пока не получим набор групп с массивами, элементы которых удовлетворяли бы условиям (1) и (2).

Таким образом, получим динамическую структуру данных, реализующую разбиение элементов на группы с возможностью поиска методом Ньютона в массиве каждой группы. Данная структура требует дополнительную память для хранения указателей на группы в количестве, равном размеру массива и размера каждой группы, что является недостатком данного метода.

На языке программирования С++ вышеописанная динамическая структура данных выглядит следующим образом:

```
template <class MyType>
struct NSGroup
{
    MyType    *data;              //массив элементов
    NSGroup   **pNextGroup;   //массив указателей следующих групп
    int       size;               //количество элементов в группе
};
```

Поиск в динамической структуре данных осуществляется по шагам следующего алгоритма:

1. Выбирать рабочую группу для поиска $G_{u,k}$, где u = 0, k = 0.
2. Выполнять поиск элемента key в массиве рабочей группы методом Ньютона.
3. В случае, если искомый элемент найден в текущей группе, то вернуть его адрес, иначе необходимо выполнить поиск в подгруппе, ссылка на которую находится между двумя элементами с индексами k и k+1, где m[k]<key и key < m[k+1].
4. Если группа $G_{u+1,k}$ пустая, то элемент не найден, иначе выбираем рабочую группу для поиска $G_{u+1,k}$ и выполняем поиск элемента в массиве этой группы в соответствии с шагом 2.

Полученный алгоритм поиска в динамической структуре данных полностью основан на методе Ньютона, в соответствии с которым скорость поиска будет порядка loglogN. Скорость поиска зависит от количества сформированных групп и от количества элементов в ней. Чем больше элементов в группе, тем меньше групп и, следовательно, выше скорость поиска.

Анализ результатов тестирования метода Ньютона с построением динамической структуры данных

Проведем сравнительный анализ метода Ньютона с бинарным методом, который на практике считается одним из наиболее быстрых. Результаты тестирования методов представлены в табл. 1.

Таблица 1 – Результаты тестирования методов Ньютона и бинарного поиска для данных типа _int64

Кол-во элементов	Метод			
	Ньютона		Бинарный	
	Кол-во сравнений	Время поиска, (мс)	Кол-во сравнений	Время поиска, (мс)
1000	11,7	–	13,8 (+18 %)	–
100000	15,7	55 (+72 %)	24,0 (+53 %)	32
1000000	17,29	690 (+73 %)	28,85 (+67 %)	400
10000000	18,68	7750 (+76 %)	34,03 (+82 %)	4406

В соответствии с полученной статистикой можно утверждать, что полученная динамическая структура данных с применением метода Ньютона выполняет меньше сравнений на 70 – 80 %, чем бинарный поиск. Причем с ростом размера массива разность количества сравнений между методами растет в пользу метода Ньютона. По времени выполнения ситуация обратная – бинарный поиск быстрее, чем метод Ньютона. Это связано с тем, что метод Ньютона использует дополнительные вычисления – в частности, использует деление и умножение, что существенно снижает скорость поиска. Также негативно влияет на скорость выполнения метода Ньютона работа с динамической памятью, т.е. с группами элементов, в то время как бинарный поиск использует для хранения данных непрерывный массив.

Стоит заметить, что операции добавления и удаления элементов из полученной динамической структуры данных будут эффективнее выполняться, чем в массиве, так как количество сдвигов элементов будет проводиться значительно меньше за

счет разбиения на группы. Это полезное свойство динамической структуры данных может быть эффективно использовано при разработке и эксплуатации баз данных интеллектуальных систем.

Разработанный метод будет полезен в тех случаях, когда скорость доступа к элементу достаточно низкая или скорость операции сравнения элементов медленная. Положительным фактором метода Ньютона есть то, что он использует однонаправленный доступ к данным, что полезно при работе со списками. То есть вместо массива в группе можно использовать список для хранения данных и эффективность метода не будет падать. Также однонаправленный доступ полезен при работе с физическим носителем.

Структура данных может использоваться для хранения упорядоченных данных и работы с ними.

Выводы

В работе рассмотрена динамическая структура данных, которая позволяет использовать сильные стороны алгоритма Ньютона и избавиться от недостатков, которые не позволяли применять данный метод для любого набора данных. Результаты тестирования показали, что метод Ньютона выполняет меньше операций сравнения относительно бинарного поиска в 1,5 – 2 раза в зависимости от количества элементов. Причем с увеличением количества элементов разрыв по сравнениям растет. Но из-за дополнительных вычислений и ресурсоемких операций деления и умножения его скорость на практике уступает бинарному поиску, в связи с чем применять метод Ньютона следует в тех случаях, когда операция сравнения элементов выполняется достаточно медленно, а также когда хранение данных в непрерывных структурах невозможно или невыгодно. Вставка и удаление в полученной динамической структуре данных выполняются эффективнее, чем в массиве.

Полученную структуру данных возможно использовать для сортировки и хранения данных, работы с ними. Также имеет смысл построение структуры типа «список» на основе разработанной структуры, так как для поиска нужного элемента будет использоваться метод Ньютона, алгоритм которого использует однонаправленный принцип движения, что для работы со списками очень важно.

Скорость поиска данного метода имеет порядок loglogN.

Литература

1. Синельников С.С. Применение численных методов к задаче поиска данных в отсортированном массиве // Радіоелектронні і комп’ютерні системи. – 2007. – № 1. – С. 68-73.
2. Березин И.С., Жидков Н.П. Методы вычислений. – М.: Физматгиз. – 1962. – Т. 1. – 632 с.
3. Вержбицкий В.М. Основы численных методов. – М.: Высшая школа, 2002. – 840 с.
4. Волков Е.А. Численные методы. – М.: Наука, 1982. – 254 с.
5. Калиткин Н.Н. Численные методы. – М.: Наука, 1978. – 512 с.
6. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ. – 2-е изд. – М.: Вильямс, 2005. – 1296 с.
7. Кнут Д. Искусство программирования. – 3-е изд. – М.: Вильямс, 2005. – 720 с.
8. Седжвик Р. Фундаментальные алгоритмы на С++: Ч. 1 – 4: Анализ, структуры данных, сортировка, поиск: Пер. с англ. (3-я ред.). – Diasoft, 2001. – 687 с.
9. Страуструп Б. Язык программирования С++. – М.: Бином, 2005. – 1098 с.

С.С. Синельников, В.О. Рєзников

Розробка структури даних для пошуку методом Ньютона

Розглянуто проблеми узагальнення існуючих методів пошуку даних, застосування динамічних структур даних до задачі пошуку, застосування методу Ньютона. Показано комплексний підхід до розв’язання задачі пошуку. Розроблено метод пошуку даних, заснований на застосуванні динамічних структур даних і розбитті масиву на групи елементів.

Статья поступила в редакцию 30.11.2007.